Problem Set #1

Greg Plaxton

January 28, 2019

Your solutions are required to be typeset using LaTeX, and to be turned in electronically using Canvas. As discussed on the course syllabus, there are two deadlines for this homework. The first deadline is 8pm on Friday, February 8. The second deadline is 8pm on Sunday, February 10. Submissions received before the first deadline will not be charged any slip days. No submissions will be allowed after the second deadline. See the syllabus for further details regarding the slip day policy.

1. Let us define the *external path length* of a binary tree as the sum of the depths of all of its leaves. (The root is at depth 0, the children of the root are at depth 1, the grandchildren of the root are at depth 2, et cetera.)

   (a) Prove that any ℓ-leaf binary tree has external path length $\Omega(\ell \lg \ell)$.

   (b) Use the result of part (a) to prove that the average-case complexity of any comparison-based sorting algorithm is $\Omega(n \lg n)$.

2. Let $h(n)$ denote $\sqrt{n}$, and let n_0 denote the least positive integer such that $h(n) < n/2$ for all $n \geq n_0$. Consider the recurrence

$$T(n) = \begin{cases} n^2 & \text{if } 1 \leq n < n_0 \\ 2T\left(\left\lfloor \frac{n}{2} + h(n) \right\rfloor\right) + n & \text{if } n \geq n_0. \end{cases}$$

   (a) Prove that $T(n) = O(n \log n)$.

   (b) This is a bonus part worth one point only. Specify a function $f(n)$ satisfying both of the following conditions: (1) if we change the definition of $h(n)$ to $f(n)/100$, then $T(n) = O(n \log n)$; (2) if we change the definition of $h(n)$ to $100 \cdot f(n)$, then $T(n) = \omega(n \log n)$. You are not required to justify your answer.

3. Problem 30–5, page 923 of CLRS.

4. Suppose we have n tasks to be scheduled sequentially on a single resource starting at time 0. The tasks are indexed from 1 to n. Task i is characterized by three positive integers: a deadline d_i, a value v_i, and an execution requirement e_i. A schedule specifies

the order in which to execute the n tasks. The value of a schedule is defined to be the sum of the values of the tasks that are completed by their deadline.

Example: Suppose we have three tasks. Task 1 has $d_1 = 8$, $v_1 = 3$ and $e_1 = 3$. Task 2 has $d_2 = 5$, $v_2 = 7$, and $e_2 = 4$. Task 3 has $d_3 = 6$, $v_3 = 2$, and $e_3 = 5$. Consider schedule $(1, 2, 3)$. Under this schedule, task 1 completes at time $e_1 = 3$, task 2 completes at time $e_1 + e_2 = 7$, and task 3 completes at time $e_1 + e_2 + e_3 = 12$. The value of this schedule is $3 + 0 + 0 = 3$ since only task 1 meets its deadline.

(a) Assume that there is a positive constant c such that $\max_{1 \leq i \leq n} e_i \leq n^c$. Give a polynomial-time algorithm to compute the value of an optimal schedule.

(b) Assume that there is a positive constant c such that $\max_{1 \leq i \leq n} v_i \leq n^c$. Give a polynomial-time algorithm to compute the value of an optimal schedule.